

SOURCE CODE GENERATING METHOD CAPABLE OF RESTORING DELETED PART

Patent number: JP9198241
 Publication date: 1997-07-31
 Inventor: TAKAHARA KIYOSHI; FUJII KEIJI; NAGAO MASAOKI
 Applicant: HITACHI LTD
 Classification:
 - international: G06F9/06; G06F9/06
 - european:
 Application number: JP19960007138 19960119
 Priority number(s):

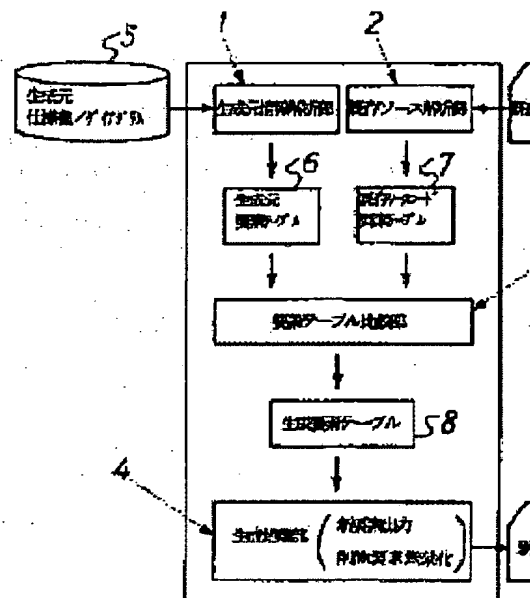
22390 U.S. PTO
 10/761207



Abstract of JP9198241

PROBLEM TO BE SOLVED: To make it possible to restore a temporarily deleted element by inputting the information unit of a corresponding generation destination to a comment sentence or a generation source at the time of generation processing after deleting a generation element.

SOLUTION: A generation source information analyzing part 1 analyzes a generation source file 5 and outputs an element to be generated to a generation source table 6. An existing source analyzing part 2 analyzes an existing source code and outputs an existing source code element table 7 indicating an element name and the necessity/unnecessity of comment preparation. An element table comparing part 3 merges both the tables 6, 7 and outputs a generation element table 8 indicating the element name and the existence of a generation source element. A generation processing part 4 generates a new source code for an element corresponding to a constitutional element inputted again on a diagram so as to return a comment to an execution sentence in accordance with the contents of the table 8. Since it is unnecessary to input deleted data again, input load can be reduced.



THIS PAGE BLANK (USPTO)

特開平9-198241
(43) 公開日 平成9年(1997)7月31日

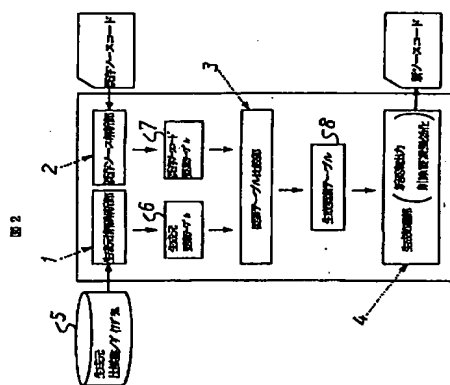
(51) Int. Cl.	識別記号	F I	特許請求 未請求 請求項の数 1 O L (金7頁)
G06F 9/06	530	G06F 9/06	530 V
	540		540 C
(21) 出願 号	特開平8-7138	(71) 出願人	000005108 株式会社日立製作所 東京都千代田区神田駿河台四丁目6番地 高原 清
(22) 出願日	平成8年(1996)1月19日	(72) 発明者	神奈川県横浜市中区戸塚区戸塚町5030番地 株 式会社日立製作所ソフトウェア開発本部内 藤井 啓明
		(72) 発明者	神奈川県横浜市中区戸塚区戸塚町5030番地 株 式会社日立製作所ソフトウェア開発本部内 長尾 雅明
		(74) 代理人	小川 勝男 井理士

(54) 【発明の名称】 削除部分の復元可能なソースコード生成方法

(57) 【要約】

【課題】 仕様 / ダイアグラム等からソースコード等
を生成する装置において、仕様 / ダイアグラム中から
削除された要素および同要素に属する部分が、既存の
ソースコードに存在する場合に、当該部分をコメント化
する等の方法でプログラムの内容に影響を加えない状態
で保存することにより、再度同一要素が追加された場合
に既存コードの復元を可能とする。

【解決手段】 仕様書 / ダイアグラムに含まれる要素を
解析する「生成元情報解析部 1」と、既存ソースコード中
の要素を解析する「既存ソースコード解析部 2」、これらの解析
結果をもとにソースコードを生成する「生成処理部 4」と
から成る。



【特許請求の範囲】

【請求項1】 電子計算機を用い、仕様書及びダイアグラムからプログラムのソースコードを生成する方法において、生成元から削除された情報に相当するコードが生成先へ存在する場合に、該当するコードをプログラムに影響を与えない形に変えて保存し、生成元で削除された情報が再入力された場合には、生成元の削除情報に基づいて、生成元で削除された情報を復元可能なソースコードの生成方法。

【発明の詳細な説明】

【0001】
【発明の属する技術分野】 本発明は、計算機支援ソフトウェア工学の分野で、計算機を用いて各種仕様書、ダイアグラムを作成し、これらからソースコード等のプログラムを生成する方法に関する。

【0002】

【従来の技術】 従来、仕様書 / ダイアグラム等からソースコード等を生成する方法では、生成元の仕様書 / ダイアグラム等から、生成要素となる情報を削除した場合、既に生成先へ対応するソースコード等がある場合でも、生成元の情報を削除して、この種の方式の例として、ISWB3 オブジェクト指向分析・設計支援 (日立製作所) がある。

【0003】

【発明が解決しようとする課題】 上記の従来技術では、仕様書 / ダイアグラム等から生成要素を削除した場合、いったん削除された生成要素が再び記述された場合などについて配慮がされていないので、実行前段階で、不意で生成元の仕様書 / ダイアグラム中の生成要素を削除した場合、生成元の対応する情報が失われるという問題があった。特に、生成した後にソースコードの中に、使用者がコーディングを追加する(ユーザ・アクション・コーディング)ような場合には、この部分も同時に削除され、その回復が困難であった。本発明は、生成要素が削除された後の生成処理時に、対応する生成元の情報単位をコメント文や生成元に取り込むことにより、生成元のプログラムの処理内容に影響を与えない形で保持することにより、再度同一の生成単位が仕様書 / ダイアグラムに入力された場合に生成時に参照されていた削除情報を復元することを目的とする。

【0004】

【課題を解決するための手段】 上記の目的を達成するため、生成情報削除後の生成時に、生成されたプログラムの内容に影響を加えない形で、削除部分に相当するソースコードを保持する機構 (図2の4等) を設けたものである。さらに、生成元で削除情報に同じ情報が入力された場合、その後の生成処理において、保持されている削除部分に対応するソースコードの復元を行う処理を加えたものである (図4参照)。

【0005】

【発明の実施の形態】 本発明においては、電子計算機等からなるソースコード生成システムにて、生成元の仕様書 / ダイアグラム・ファイルを読み出し、生成すべき要素 (生成要素) を出力する。一方、既存ソースコードを解析し、要素名及びコメント化の有無 (既存ソースコード要素) を出力する。さらに、これらをマージ処理し、要素名、生成要素の有無、及び既存ソースコード要素の有無 (生成要素) を出力する。前記システム上の生成処理は、その生成要素に基づき、ダイアグラム上で再度入力された構成要素に相当する要素について、コメントから実行文に直し、新しいソースコードを生成する。なお、削除情報のコメント化の他に、その削除情報に相当する生成元の情報を、ソースコード生成時に生成元の仕様書 / ダイアグラム・ファイルに取り込み保存してもよい。また、生成元の仕様書 / ダイアグラムを編集中に、削除情報に合致する情報が新たに入力された場合、削除されていた情報仕様書 / ダイアグラム上に復元してもよい。

【0006】 以下、本発明の一実施例を図面により詳細に説明する。図1は、本発明の一実施例におけるソースコード生成方法の作業手順 (プログラム作成工程) を示す図であって、①～⑥は作業手順である。

①ダイアグラムエディタによりダイアグラムを新規に作成する。

本実施例では、手順①に示すダイアグラムエディタの画面中、A、B、Cはダイアグラムでキーとなる項目であり、プログラム生成においてプログラムの構成要素 (例: C++言語に置けるクラス) となるものである。

②ダイアグラムからプログラムを生成する。

本実施例では、手順②で示したダイアグラムから図中のA、B、Cに対応して、Class A、Class B、Class Cの各宣言文が出力される。

③ダイアグラムを再度編集し、Cを削除する。

④再度ダイアグラムからプログラムを生成する。この時、手順③で削除されたCに相当するClass Cとそれに付随するプログラムコードがコメント文に置き換えられる。これにより、これらのコードは、プログラムの実行に影響を与えない形で、プログラム中に保存される。⑤さらに手順④のダイアグラムを編集し、手順③で削除したCと同名の要素を再度作成する。

⑥手順⑤の状態で、プログラムを生成する。

この時、手順⑤でコメントとなったClass Cに関するコードが、再び有効なコードとして復元する。【0007】 ここで、本実施例のソースコード生成方法を行うためのシステム構成を図2に示す。生成元情報解析部1は、生成元となるダイアグラム (生成元ファイル5) を解析し、生成すべき要素 (図1に示したA、B、C) を生成要素テーブル6に出力する。生成要素テーブル6の構成項目は、例えば図3の(a)に示すように、【要素名】 (図1に示した構成要素を一意に特定する

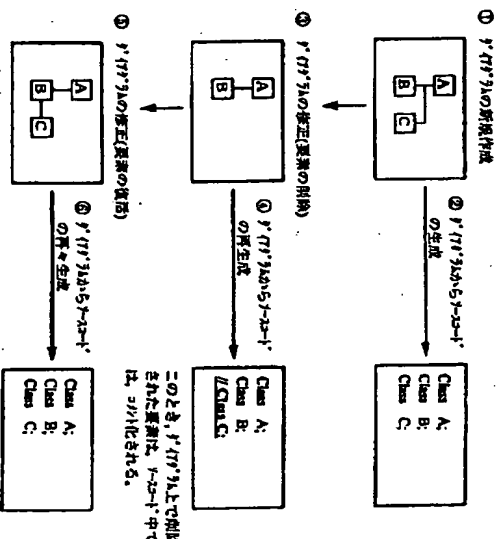
名称)である。同様に既存ソース解析部2は既存ソースコードを解析し、既存ソースコード要素テーブル7を出力する。既存ソースコード要素テーブル7の構成項目は、例えば図3の(b)に示すように、[要素名、コメント化の有無]である。要素テーブル比較部3は、生成要素テーブル6と既存ソースコード要素テーブル7をマージし、生成要素テーブル8を出力する。この生成要素テーブル8の構成要素は、例えば図3の(c)に示すように、[要素名、生成元要素テーブル中での要素の有無、既存ソースコード要素テーブル中での要素の有無、既存ソースコード要素テーブル7のコメント化の有無]である。なお、図3は生成処理部4による生成時点での各テーブルの内容を示す。生成処理部4は、生成要素テーブル8に従い、ダイアグラム上で再入力された構成要素に相当する要素について、コメントから実行文に戻すよう、新しいソースコードを生成する。

【0008】ここで、生成処理部4の処理を図4に示す。生成処理部4は、生成要素テーブル8の第1行から最終行まで、各要素名について各項目をチェックし(ステップ401)、生成元要素テーブル6及び既存ソースコードテーブル7中の有無、コメント化の有無が、それぞれ「有、有/無」であれば、その要素をコメント化して出力する(ステップ402)。なお、図3の(c)に示した例ではCが該当する。また、それぞれ「有、有/無、有/無」であれば、その要素を新規に出力する(ステップ403)。なお、図3の(c)に示した例では該当要素は存在しない。また、それぞれ「有、有、無」であれば、その要素を出力する(ステップ404)。なお、図3の(c)に示した例ではA、Bが該当する。また、それぞれ「有、有」であれば、コメント化されている要素を有効化して出力する(ステップ405)。なお、図3の(c)に示した例では該当要素は存在しない。

【0009】前記実施例の他に、様々な応用が考えられる。例えば、図1の手順④において削除されるべきコードをコメント化したプログラム中に保持し、手順⑤においてコメント化されたコードを復元するようにしてもよい。すなわち、前記実施例と同様の機能で、保存するソースコードをコメント化する方法は無く、ダイアグラム側に盛り込んで保存する方法である。より具体的には、図2に示した生成処理部4において、[削除要素無効化]をソースコードをコメント化する方法は無く、削除

【図1】

図1



すべきソースコードとその位置を生成元情報(図2の別では、生成元は修飾/ダイアグラム・ファイル5)の中へ保存する方法が挙げられる。

【0010】さらに、本実施例及び前記方法を可能にするような編集装置を構成してもよい。すなわち、削除された構成要素に関する情報を保持すること可能とし、再度入力した構成要素が以前に削除された構成要素と合致するか判断し、合致した場合には削除データとして保持していたデータを復元するものである。より具体的には、前述のダイアログの場合では、ダイアログのシンボルを特定する項目をキー項目とし、削除時にキー項目と付随する項目をリスト等の形式で保存し、新規に入力されたキー項目と一致する場合に、保存されていた付随項目を復元する方法がある。なお、本実施例の応用として、削除された項目と一致した場合に、既存項目を復元させるかを利用者に選択可能とするユーザインタフェースを用意する方法や、削除されたキー項目、もしくは付随項目の一覧を画面等に表示し、ユーザが選択可能にする方法が挙げられる。

【0011】

【発明の効果】本発明によれば、仕様がダイアグラムの記述とソースコードの生成を繰り返した場合でも、いったん削除した要素が復元されるので、削除したデータを再度入力する必要がなく、入力の負担が軽減され作業効率が向上する。また、使用者の不注意によって仕様書/ダイアグラムに変更を加えてソースコードを生成した場合にも、復元が可能となる。

【図面の簡単な説明】

【図1】本発明の一実施例におけるソースコード生成方法の作業手順を示す図である。

【図2】本発明の一実施例におけるソースコード生成システムの構成要素を示す概略図である。

【図3】本発明の一実施例における生成要素管理用の各テーブルを示す図である。

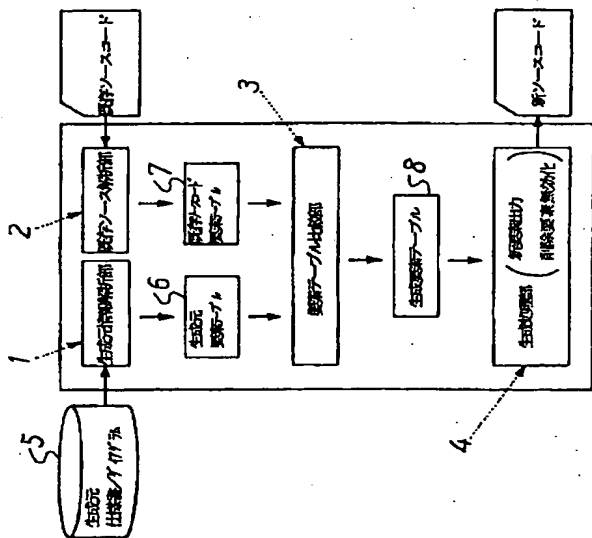
【図4】本発明の一実施例における生成処理部の処理方法を示す図である。

【符号の説明】

- 1: 生成元情報解析部、2: 既存ソース解析部、3: 要素テーブル比較部、4: 生成処理部、5: 生成元は修飾/ダイアグラム・ファイル、6: 生成元要素テーブル、7: 既存ソースコード要素テーブル、8: 生成要素テーブル。

【図2】

図2



【図3】

図3

(a) 生成元変換テーブル

変換名
A
B

(b) 既存ソースコード変換テーブル

変換名	コメント化の有無
A	無
B	無
C	無

(c) 生成変換テーブル

変換名	生成元テーブル中の有無	既存ソースコード中の有無	コメント化の有無
A	有	有	無
B	有	有	無
C	無	有	有

(7)

特開平9-198241

【図4】

図 4

